CodeSurfer—Product Overview

GrammaTech, Inc., February 28, 2005

What it is: *CodeSurfer* is a source-code browser that understands pointers, indirect function calls, macros, and whole-program effects. CodeSurfer also uses model-checking on the control-flow graph to check sequencing properties of programs. *CodeSurfer for C/C++* is a commercial tool that is available from GrammaTech, Inc. Typical uses include reverse engineering, debugging, safety/security auditing, and documentation.

Features: CodeSurfer offers many features, including:

- **Navigation.** Navigate …
  - From a #include directive to the included file.
  - From a macro use to a macro definition.
  - From a type use to the type definition.
  - From a direct function call site to the function definition.
  - From an indirect call site to the function definition(s).
  - From a variable occurrence to the variable declaration.
  - From a variable use to the statements that account for its value.
  - From a variable assignment to the statements that use the assigned value.
  - From a statement to the control points that affect whether the statement gets executed.
  - From a control point to the statements whose execution depend on it.
- **Program slicing.** This feature highlights code relevant to understanding a particular issue and does impact analysis.
- **Pointer analysis.** This feature tracks loads and stores via pointers. The targets of indirect function calls are also calculated.
- **Automation of common tasks.** For example,
  - Trace the flow of data backward and forward through code
  - Display what variables a pointer can point to
  - Highlight code that affects selected statement(s) and/or variable(s)
  - Display the call graph, including calls through function pointers
  - Determine the impact of possible code changes
  - Extract detailed program information for documentation
- **Model checking.** This feature examines the control-flow graph to check sequencing properties of a program.
  - It answers complex questions about the flow of execution.
  - Queries are constructed with templates. For example, one query checks that *A* always occurs between *B* and *C*, where *A*, *B*, and *C* are user-defined sets of program points.
  - Example of how model-checking can be applied**:** say that your application uses a DNS library, and the library contains an initialization function called `initialize_dns`. Before calling any other routines in the library, a program must call `initialize_dns`. You wonder if it is always true that `initialize_dns` is called before the other routines. Instead of manually wading through the code to answer this question, you can ask the CodeSurfer Path Inspector. The Path Inspector will either tell

you that `initialize_dns` is always called first, or it will show you a counterexample—an execution path of the program that calls one of the other functions in the DNS library without first calling `initialize_dns`.
    - Note: the model checker can return false positives (that is, report something as an issue when it isn't) and can miss potential issues, so potential users are encouraged to discuss their application with GrammaTech to ensure that they understand the technology's limitations.
- **API for customization and batch processing.** CodeSurfer's scripting language, which provides access to the dependence graph program representation, can be used to build batch program-analysis applications, or to integrate CodeSurfer with other tools. The Core API consists of 19 libraries that contain 322 fully-documented functions. The following program representations (among others) are calculated:
    - Abstract Syntax Trees (ASTs) [with a pattern matching API]
    - Preprocessor Expansions and Include Trees
    - Points-To and Pointed-To-By Sets
    - Control Flow Graphs (CFGs)
    - Call Multi-Graphs [direct & indirect]
    - Def, Use, and Conditional-Kill Sets [per statement]
    - Non-Local Def and Use Sets [per procedure]
    - Control and Data Dependences (PDGs) [per statement]
    - Transitive in/out data dependences [per procedure]

A more complete list of features is available at http://www.grammatech.com/products/codesurfer/overview.html and http://www.grammatech.com/products/codesurfer/overview_pi.html

Benefits: CodeSurfer makes it easy to understand and analyze code. For example, some users have inherited legacy programs that they must modify or maintain. Others use CodeSurfer for software inspections. Inspection time is usually reduced substantially. For example, the following quotation is from a user:

> *We use a prototype static analyzer from a university for datarace detection. The output must be processed "by hand" in order to locate the root cause: this is quite a hard job! CodeSurfer is used as an assistant for the user in his task of locating the root cause of a detected datarace condition. Datarace conditions are due to misuse of POSIX mutexes…In the absence of an assistant tool, the "manual" analyses require about 2 to 5 days full time for one person. When using CodeSurfer, the same task has been reduced to 2 hours.*

Contexts in which it is best used: CodeSurfer works best on projects where most of the code is written in C/C++. The tool has been tested with code written for many popular C/C++ compilers (e.g., GNU/gcc and the Microsoft compilers). The most detailed analyses are available if the program being examined is 200,000 lines of code or smaller, but many users have used CodeSurfer on larger projects. Setting up CodeSurfer to work with your project is similar to setting up a compiler, and CodeSurfer can use most makefiles.

What will a collaboration look like? Prior to writing a collaboration proposal, you should communicate with us to determine whether your application is a good fit for CodeSurfer. GrammaTech will work with you during proposal development on planning your collaboration. In addition, we should jointly determine what metrics would make sense to collect. We are also happy to help you with writing the proposal, if you feel that is helpful. At the start of the collaboration, we suggest a 2-day formal training course at your site, based on our training materials but tailored to your needs. Alternatively, we can do some training over the web (using WebEx). A tutorial is available, as well. We will help you set up CodeSurfer on your project. We will provide you with unlimited customer support via telephone and email (as we do for all our customers).